


**IUT de VANNES**
**Département Informatique**
**Bp 561**
**56017 VANNES CEDEX**

**Adrien Bonnaire**

**Groupe C**

Enseignant responsable :

**M. Gilles Gaffiot**




# CONCEPTION D'UNE APPLICATION WEB POUR LA GESTION DE FICHES ETUDES



Stage effectué du 17 avril au 23 juin 2006 en vue de l'obtention du DUT Informatique, sous la responsabilité de :
M. Jean-Hugues Blanc, directeur d'ADRIA SENSO

Au sein de l'Entreprise : **ADRIA**
**Creac'h Gwen**
**F-29196 Quimper Cédex**

**Le 16 Juin 2006**

# Remerciements

Je tiens à préciser que ce stage s'est déroulé dans des conditions admirables, au sein d'un projet ambitieux et surtout encadré par des personnes aussi sympathiques que professionnelles.

C'est pourquoi, je veux remercier tout particulièrement mon responsable de stage Monsieur Jean-Hugues Blanc pour m'avoir fait confiance et pour m'avoir bien encadré durant le stage.

Je remercie Monsieur Mickaël Aucouturier, Nicolas et David qui grâce à leurs compétences techniques m'ont permis de mener à bien ce stage.

Je remercie également tout le personnel de l'Adria Senso avec en particulier Mesdames J. Cavalazzi, M. Galesne et Monsieur A. Marty pour leur gentillesse et leur bon accueil.

Mes remerciements vont enfin vers Monsieur Gilles Gaffiot, mon tuteur de stage de l'IUT de Vannes, pour ses enseignements ainsi que Monsieur François Morice pour m'avoir aidé à trouver ce stage.

Je pense également à Thibault Le Reste, étudiant stagiaire de l'Iut de Vannes et collègue durant ce stage, pour m'avoir conduit tous les jours jusqu'à l'Adria et pour son travail effectué sérieusement.







# Sommaire



# Introduction

La société Adria Senso réalise des tests sensoriels sur divers produits du secteur agroalimentaire pour des clients industriels, distributeurs etc. Cette société est présente à Quimper, Paris, Lyon, Toulouse et Clermont-Ferrand.

Chaque étude répond à un cahier des charges précis. Ces études sont enregistrées par des responsables panels dans un logiciel qui s'appelle Siges (Système informatique de gestion des études sensorielles). Aujourd'hui, ce logiciel n'est plus maintenable et est trop ancien. De plus, l'Adria Senso s'est développée ce qui fait qu'il ne répond plus complètement à l'utilisation souhaitée par les utilisateurs de l'entreprise.

Comment faire pour que le redéveloppement de l'application Siges soit plus rapide, plus pratique et plus économique ?
Est-t-il possible de redévelopper l'application Siges en s'orientant vers une technologie intranet type architecture 3-tiers ?

Pour mener à bien ce stage de deuxième année du 17 mai au 23 juin 2006, je travaillerais, sur le site de Quimper, en binôme avec la collaboration de Thibault Le Reste. J'espère pouvoir mettre en pratique ou apprendre des savoirs tant en réseau (légèrement), qu'en programmation ou en génie logiciel etc. De plus, j'ai enfin l'opportunité de m'épanouir dans un environnement professionnel donc pourquoi ne pas s'en priver.

Dans ce rapport vous trouverez quatre parties distinctes :

- la première partie présente le contexte du stage,
- la deuxième s'occupe de décrire la démarche à suivre pour la résolution du problème,
- la troisième donne, tout en analysant, les grandes lignes de la résolution du problème,
- la dernière affiche les résultats obtenus et analyse les écarts par rapport aux prévisions.







# Chapitre 1

## L'environnement du stage au sein de l'Adria Senso : contexte et mission de départ





# 1. Présentation du cadre du stage pour mieux situer le projet

Cette partie va décrire le cadre du stage. Elle va simplement donner les éléments nécessaires pour mieux comprendre le contexte de mon travail.

## 1.1. Le groupe ADRIA

L'ADRIA a été fondé en 1971 sous une forme associative. Son rôle est d'accompagner les entreprises du secteur agroalimentaire dans leur processus d'innovation, dans la maîtrise de la qualité et dans la formation du personnel.




***Fig. 1.1 :*** *logo de l'Adria*

Les principaux concurrents de l'Adria sont EuroFins, Silliker et Alcontrol.

Liste des sociétés en concurrence :

| | |
|---|---|
| Eurofins | Silliker |
| Axans | Alcontrol |
| Cerbia | BVA |
| ISHA | Secodip |
| Repère top 10 | Stratégir Bordeaux |
| Sam France | Siliker Bio |
| Sensorum | Maisons du gt |
| Eurosyn | IFF |
| Repère top 10 | ITG |
| Sam France | Strasbourg local |
| CHI2 | Aérial |
| TNS | Enil |
| Cécodip | IFBM, |
| Sofres | Qual Techr |
| | Agrotec Agen |
| | CTCPA, subvention |
| | Agro-analyse |
| | Iterg |
| | GT et couleur de Limoges |

Comme toutes entreprises ambitieuses, l'ADRIA a développé ses compétences. En juillet 2003, l'association s'est scindée en trois entités :

- Adria Développement (recherche microbiologique, développement produits et procédés, formation professionnelle et conseil, réglementation)
- Adria Laboratoires (analyses microbiologiques et chimiques)
- Adria Senso (évaluations sensorielles)

Il faut noter que j'ai effectué mon stage dans les locaux de l'ADRIA pour l'ADRIA Senso.





# 1.2. L'ADRIA Senso

## 1.2.1. Quelle est la mission de l'ADRIA Senso ?

ADRIA SENSO est une société spécialisée dans l'évaluation et la caractérisation sensorielle. Elle réalise des prestations de tests consommateurs, de profils sensoriels et de la formation aux techniques sensorielles pour les clients industriels, distributeurs, groupement qualité, organismes certificateurs, fédérations, restaurateurs… sur des gammes de produits alimentaires (des biscuits au pet-food en passant par les produits surgelés, frais) et hors alimentaires tels que la cosmétique (crème, démaquillants, savon, shampoing…) ou la parapharmacie (dentifrice…).

## 1.2.2. Fiche d'identité de l'ADRIA Senso

Nom de l'entreprise : Adria Senso

Raison sociale : SAS (Société par Actions Simplifiée)
Tableau explicatif du statut juridique de type SAS :

| **Capital** (minimum en euros) | **Associés** (nombre minimum) | **Qualité des associés** | **Objet** | **Responsabilité des associés** | **Régime fiscal** |
|---|---|---|---|---|---|
| 37 000 | 2 | Personne morale ou physique | Commercial | Limitée aux droits sur l'actif social | Impôt société |

Adresse et siège social : Adria Creac'h Gwen F-29196 Quimper Cedex

Téléphone : 02.98.10.18.10
Fax : 02.98.10.24.47

Site Internet : http://www.adria.tm.fr/

Secteur d'activité : Agroalimentaire
Chiffre d'affaires : 2.200.000€

Quelques chiffres :
- Une équipe d'une vingtaine de personnes.
- Plus de 1200 tests par an
- Plus de 45000 consommateurs qui ont goûté un produit
- 2 salles de dégustation à Quimper et à Toulouse de 20 à 15 boxes qui sont conformes à la norme XP V 09-500, informatisées et équipées d'une cuisine « type cuisine collective » (matériel de cuisson, plaques, four, frigo, positif et négatif).





## 1.3. La place et le rôle de l'informatique dans l'entreprise

Autrefois, les traitements des études s'effectuaient uniquement sur papier, à la main. Ce travail était fastidieux pour l'équipe de l'Adria Senso.
Aujourd'hui, ce travail est facilité par l'informatique. L'entreprise possède quasiment dans chaque salle un ou plusieurs ordinateurs reliés par un réseau intranet.

Dans l'entreprise les employés de l'Adria Senso, par exemple les chargés de panel, utilisent un logiciel nommé Siges à Quimper ou Léoni à Toulouse. Ce logiciel leur permet d'enregistrer les données concernant chaque étude (cahier des charges de chaque étude).

On a vu (au point 1.2.2) que les salles de dégustations sont informatisées. En fait, chaque consommateur qui vient goûter les produits se place dans un boxe et dispose d'un ordinateur. Le consommateur évalue les produits et rentre les résultats directement dans un logiciel appelé FIZZ. Les résultats peuvent donc être édités très rapidement.

On peut ajouter que chaque partie de l'entreprise est protégée par un système de portes électronique. Pour franchir les portes, il faut donc disposer d'une carte avec code barre.

# 2. Objectifs du stage

## 2.1. Les motivations de l'entreprise

L'entreprise dispose d'un logiciel qui permet d'enregistrer, de modifier ou d'afficher des informations relatives à chaque étude. Le logiciel est appelé différemment que l'on soit à Quimper ou Toulouse. Cela ne change rien du tout puisqu'il s'agit de la même application. A Toulouse, le nom de l'application est Léoni alors qu'à Quimper c'est Siges (prononcé Si-Ji-S). En pratique, les employés partent du cahier des charges d'une étude et l'enregistrent dans le logiciel. Par la suite, il est possible d'ajouter de nouvelles informations à cette étude. Ils peuvent aussi afficher les informations qui les intéressent telles que la description du test d'un produit de l'étude, la facturation de l'étude etc.

Le logiciel SIGES est programmé en Delphi. Il doit être chargé sur chaque machine. Ce logiciel a été développé par l'entreprise Team Partner. Celle-ci n'existant plus aujourd'hui, il n'y a plus de maintenance possible à l'exception de M. Mickaël Aucouturier qui est informaticien au sein de l'Adria Senso et qui connaît le logiciel.

Actuellement, l'application demande beaucoup de ressources et monopolise une grande partie de la bande passante pour les échanges de données entre les différents sites de l'ADRIA SENSO. Une ligne SDSL (Symetric Digital Subscriber Line) est utilisée mais elle coûte très cher à l'entreprise.

De plus, les données sont stockées dans une base de données Oracle. Le problème est qu'il y a en réalité deux bases de données distinctes. Toulouse et Quimper ont accès à l'une et Paris, Lyon, Clermont etc. à une autre. Mais ces deux pôles ne sont pas reliés. Par exemple, Lyon ne sait pas ce que Toulouse étudie.

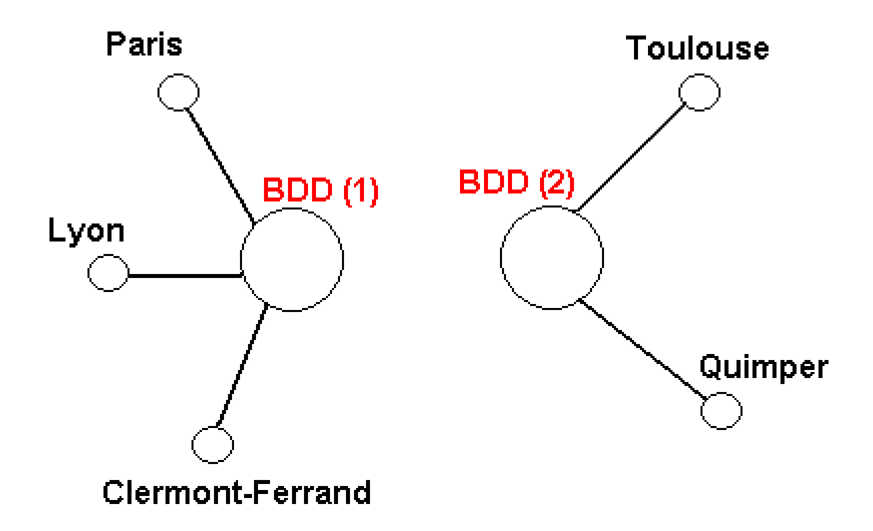



***Fig. 1.2 :*** *organisation actuelle des données*

L'entreprise aimerait qu'il n'existe qu'une seule et unique base de données.

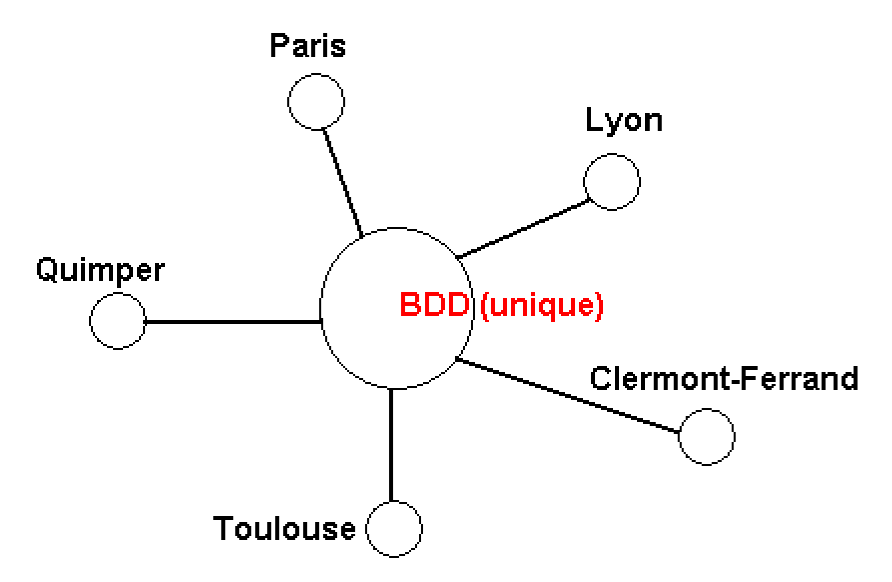



***Fig. 1.3 :*** *organisation finale des données souhaitée*

**Aujourd'hui, l'entreprise cherche donc un nouveau moyen qui remplacerait le logiciel Siges et qui serait plus rapide, plus pratique et plus économique.**





## 2.2. Cahier des charges

### 2.2.1. Analyse de l'existant

Les documents suivants sont disponibles :

- Le document d'analyse (version 1) de l'application LEONI
- Le manuel utilisateur (version 1) de l'application LEONI V2
- Le document de description des tables de la base de données ainsi que les modèles conceptuels attachés, de l'application LEONI (en Mérise).

Le problème de ces documents c'est qu'ils ne correspondent pas forcément à la dernière version ou qu'ils ne sont pas assez détaillés.

### 2.2.2. Qui seront les utilisateurs de l'application ?

Cette application est destinée aux employés de l'ADRIA SENSO et plus particulièrement les chargés de Panel (CP). Ils sont répartis entre Quimper et Toulouse.
Il y a des droits différents en fonction de la hiérarchie des employés. Exemple : l'administrateur possède tous les droits.

### 2.2.3. Objectifs fixés par le stage : la mission, les résultats attendues

L'objectif principal est de reproduire une partie de l'application Siges. M. Jean-Hugues Blanc a définit ses principales attentes dés le début du stage pour les 10 semaines suivantes. Il faut donc reproduire la partie du logiciel qui s'occupe du cahier des charges des fiches études. Les fonctions et les particularités suivantes sont donc attendues :

- **Lister des fiches études**

Cette rubrique permettra d'afficher la liste de toutes les fiches études. L'utilisateur peut changer l'affichage comme il le souhaite et passer de l'affichage de toutes les fiches études à l'affichage par pages et vis versa.
La liste fournira les principales informations de chaque étude pour faciliter les recherches : son numéro, son type (étude, baromètre, autre …), son intitulé, son objet, son contrat, sa date de création et son stade (A, C …).
Il devra être possible de sélectionner une fiche pour en avoir ses informations ou pour la modifier. En revanche nous nous intéresserons qu'à la modification. Il n'est pas prévu de créer une nouvelle fiche.

- **Rechercher des fiches études**

Dans la page qui affiche les liste des fiches études, l'utilisateur doit pouvoir effectuer des recherches de fiches à l'aide de différentes requêtes (numéro, délais, typologie, date…).

- **Afficher les informations d'une fiche étude**

La fiche étude sera composée de cinq fonctions principales :

**Caractéristiques générales**
Contient les caractéristiques générales d'une étude.

**Tiers**
Contient la liste des tiers demandeurs de l'étude. Chaque tiers demandeurs possède un ou plusieurs tiers destinataires et facturés. Concernant les tiers facturés, l'utilisateur peut ouvrir une autre fenêtre (en cliquant sur un lien ou un bouton) qui contient tous les éléments de facturation.

**Produits**
Contient la liste des produits de l'étude avec leurs éventuels souscripteurs associés.

**Tests consommateurs**
Contient la liste des tests consommateurs (sous-test - conditions) de l'étude et la liste des traitements du test consommateurs. Concernant les conditions de test, l'utilisateur peut ouvrir une autre fenêtre (en cliquant sur un lien ou un bouton) qui va détailler ce sous-test.
Cette sous fenêtre possède d'autres onglets : lieux de test, description du test et questionnaires, protocole, cibles et traitements.

**Analyses**
Contient la liste des analyses (mesures) et la liste des traitements de l'analyse pour l'étude en cours.

Remarque : dans chaque onglet ou sous fenêtre, l'utilisateur doit pouvoir modifier des informations et les enregistrer.

- **Imprimer**

A tout moment, l'utilisateur peut imprimer la liste des fiches études qui apparaissent à l'écran. Après avoir sélectionné une fiche, il pourra imprimer son détail.

- **L'interface simple**

L'application devra ressembler à l'application déjà existante (SIGES). Néanmoins, des améliorations graphiques pourront être développées.
Par exemple l'interface pourra comporter des onglets ou des menus qui correspondront aux fonctions de la fiche étude.
On optera pour une interface simple et modulable.
L'utilisateur devra facilement savoir où il se trouve dans l'application et ne devra pas se perdre dans les différents menus.

- **Configurer l'application**

Des modules de configuration pourront être développés.
Exemple : Attribution des droits des utilisateurs, configuration de l'interface, etc.

## 2.3. Création de l'application : des contraintes techniques à définir

L'application développée doit être compatible avec Windows 2000 Professionnel NT et supérieur et fonctionner sous Internet Explorer 6 et supérieur. Les écrans d'ordinateurs dans l'Adria sont principalement de taille 15 pouces configurés en définition 1024 par 768 pixels. Il ne faut pas oublier de prendre en compte cette contrainte.

Les salariés de l'Adria Senso utilisent tous les jours le logiciel Siges. Ils ont par conséquent besoin de la base de données Oracle. Celle-ci contient une soixantaine de tables telles que : Analyses, Produits, Etudes etc. C'est une base très lourde, par exemple, la table Produits contient environ 15000 tuples (lignes).
Lors du développement de l'application cette base ne devra pas être modifiée. Pour les tests, il est donc important de travailler sur une base indépendante de celle existante. Pour cela, il est prévu de réaliser une copie de la base.




***Fig. 1.4 :*** *logo d'Oracle*

Une étude doit être réalisée pour connaître la meilleure méthode de développement de l'application (langage de programmation); Exemple : PHP5, ASP.NET etc.



# Chapitre 2

## La démarche à suivre pour la résolution du problème







Cette partie reprend le travail effectué lors de la phase d'analyse. Le document de spécification est disponible dans sa version complète en annexe. Le principal but de cette partie est de comprendre, sans rentrer dans des détails trop techniques, comment il va falloir résoudre le problème.

# 1. Analyser le problème

## 1.1. Le point de départ de l'analyse : les cas d'utilisation

### 1.1.1. Description des acteurs

Actor

- **L'Utilisateur**

C'est l'utilisateur de l'application. Il s'agit des employés de L'ADRIA qui ont besoin de compléter ou étudier des fiches études pour effectuer les études.

- **L'imprimante**

Elle va permettre l'impression de certains documents, par exemple de la liste des fiches études.

- **Le SGBD**

Il s'agit de la base de données Oracle où toutes les informations seront stockées, par exemple les fiches études, les collaborateurs, les tiers…





### 1.1.2. Le diagramme des cas d'utilisation de l'application à développer

Diagramme synoptique de Jacobson de l'application web

Application Web

SGBD

Lister toutes les fiches études

Rechercher des fiches études de manière simple

Imprimer la liste des fiches études

Utilisateur

S'identifier

Rechercher des fiches études de manière avancée

Afficher/Modifier une fiche étude

Imprimante

Configurer l'application

Créer une fiche étude

Imprimer la fiche en cours

***Fig. 2.1 :*** *diagramme des cas d'utilisation de l'application web (UML)*

Remarque : M. Blanc nous a demandé de ne pas développer le cas « créer une fiche étude ». L'application s'intéresse principalement à la modification.





## 1.2. L'architecture globale de l'application

Le schéma suivant illustre l'architecture globale de l'application.

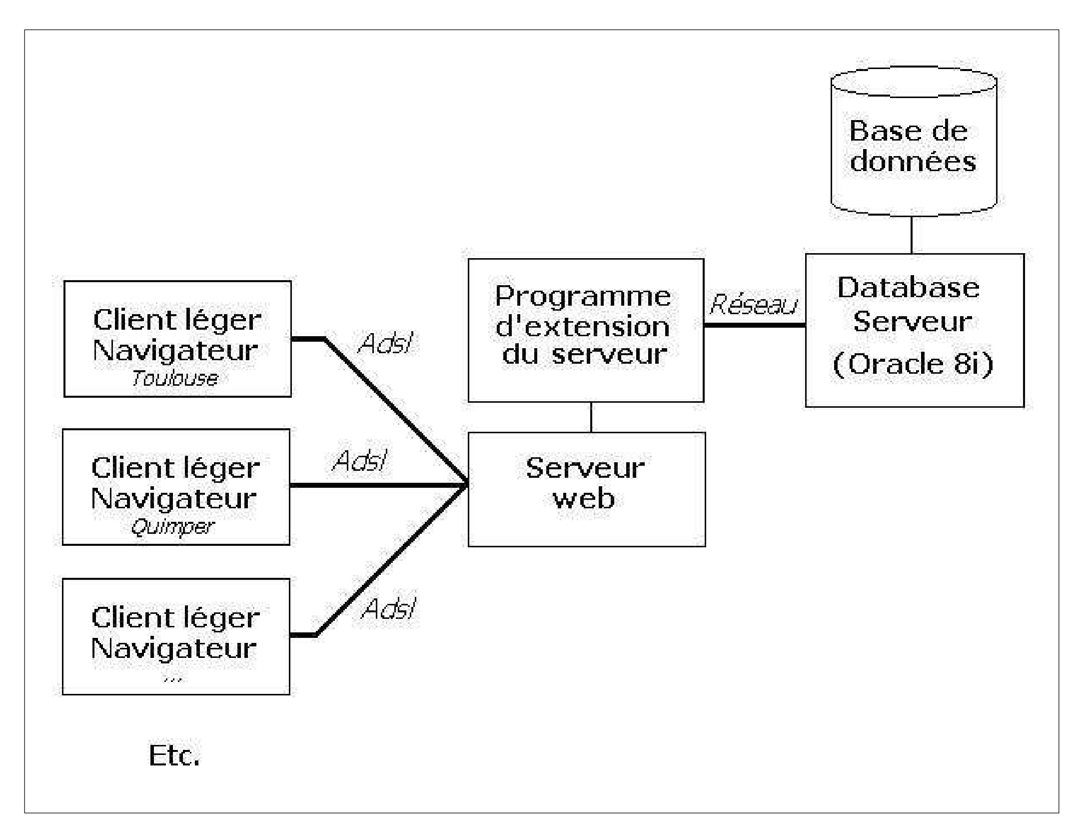



***Fig. 2.2 :** architecture globale de l'application*

C'est une **architecture 3-Tiers**, celle-ci est composée de 3 couches :

- La couche présentation (ou affichage) associée au client qui de fait est dit "léger" dans la mesure où il n'assume aucune fonction de traitement.

- La couche fonctionnelle liée au serveur, qui dans de nombreux cas est un serveur Web muni d'extensions.

- La couche de données liée au serveur de base de données (SGBD).



# 2. Faire des choix avant le développement

## 2.1. Un choix important : le langage de programmation

Le choix du langage lorsque l'on construit une application est très important car chaque langage possède des spécificités et des contraintes différentes. Quelque fois c'est le client qui fait ce choix.

Pour choisir le langage de programmation pour développer l'application Siges, nous avons rédigé un document d'analyse des langages serveurs (ANNEXE). Dans ce document nous avons listé les principaux langages serveurs avec leurs avantages et leurs inconvénients à savoir les langages PHP, ASP, ASP.NET, PYTHON, JAVA/JSP et SERVLET et PERL. Les langages serveurs sont des langages qui permettent de construire des applications qui peuvent être lues avec un simple navigateur tel qu' Internet Explorer ou Mozilla.
Après cette étude deux langages nous ont semblé plus intéressant pour le développement de Siges : PHP et ASP.NET. Nous les avons donc comparés. Pour résumer PHP5 est un langage orienté objet, qui possède tous les avantages d'ASP.NET mais qui se distingue par sa gratuité.

**Au final, notre choix s'est donc tourné vers PHP dans sa version 5.**




***Fig. 2.3 :*** *logo de PHP version 5*





## 2.2. Organiser le code : la méthode des templates

Un des reproches les plus récurrents que l'on peut faire à PHP est que le code est souvent mal organisé et qu'il devient vite illisible. En revanche, il est possible de développer rigoureusement les applications en PHP en réalisant les pages sous forme de templates et en séparant le code métier du code SGBD.

**Nous avons donc décidé d'utiliser les templates.**

## 2.3. Quel logiciel faut-il utiliser pour interpréter les pages codées en PHP5 ?

Dans un premier temps nous pensions télécharger et installer un logiciel qui possède Apache2 et PHP5 et qui est configuré par défaut. Tout d'abord, nous avons pensé à EasyPhp mais ce logiciel permet de lire uniquement les pages codées en PHP4 ou inférieures. Nous pensions ensuite utiliser Wamp qui dispose de PHP5 mais finalement nous avons trouvé sur le net des explications détaillées pour l'installation et la configuration d'Apache2 et de PHP5.

**Pour interpréter les pages que nous allons coder, nous avons donc choisis d'installer et de configurer nous même Apache2 et PHP5.**

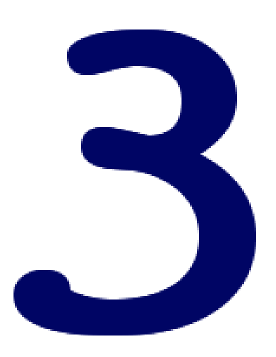



***Fig. 2.4 :*** *logo d'Apache*



# Chapitre 3

## Grandes lignes de la résolution, description et analyse des activités







# 1. Les étapes

Avant la livraison à un client, le développement d'une application passe par plusieurs étapes définies au cours des deux dernières décennies. Lors du stage, nous avons essayé de suivre ses étapes tout en prenant en compte les demandes de notre responsable de stage.

## 1.1. Durant la phase de spécification

### 1.1.1. Définir le besoin du client

Au début de chaque projet informatique, il est important de bien comprendre ce que souhaite le client et quel est le problème posé. Pour cela, il faut rédiger le cahier des charges ou le re-rédiger si le client l'a déjà fait.

Avant le début du stage Thibault et moi-même avons rencontré M. Jean-Hugues Blanc pour se présenter, lui donner nos motivations et pour lui expliquer nos qualités et compétences afin de lui démontrer qu'il va faire le bon choix dans son recrutement. Lors de cet entretien M. Jean-Hugues Blanc nous a donné un bref aperçu du sujet du stage qui nous a semblé tout de suite très intéressant.

Lorsque le stage a commencé, il a donc fallu rédiger le cahier des charges. Pour cela, dés le premier jour, après une présentation de l'entreprise et une visite des locaux de celle-ci, nous avons eu une explication détaillée (cette fois-ci) du problème et du sujet. Lorsque nous avions des doutes, nous notions nos questions puis allions voir M. Blanc pour les lui poser.

Je trouve que la définition du besoin a été plutôt facile puisqu'elle a duré 4 jours. Le résultat, très légèrement remanié pour ce rapport, se trouve ci-avant (cf. Chapitre 1).

### 1.1.2. Rédiger le document d'analyse et de conception

Cette partie est très importante puisqu'elle va nous permettre de développer l'application assez facilement lors de la phase de codage. Nous aurons à l'aide de ce dossier toutes les clés en main pour le développement. De plus, ce dossier permet de valider avec M. Blanc que ce qu'il attend va bien être développé. Enfin, il pourra être réutilisé par d'autres informaticiens s'ils doivent apporter des modifications à l'application.

Pour rédiger ce document, il a suffit de mettre en pratique les cours de Génie logiciel. Nous sommes donc partis des cas d'utilisation et des acteurs en construisant le diagramme des cas d'utilisation (use case).
Exemple de cas d'utilisation : s'identifier, lister les fiches études, etc.
A partir de là, on peut créer les diagrammes de séquences, avec leurs éventuels scénarios de remplacement, décrivant les différents cas d'utilisation. A cela s'ajoute le diagramme état-transition (state) pour définir les liaisons entre les différentes pages (objets). C'est seulement à partir de ces différents diagrammes que nous avons affiché les principales fenêtres IHM (Interface Homme Machine).

Normalement, il faut construire un ou plusieurs diagrammes de classe mais nous disposons déjà de ce diagramme (dans le document d'analyse de l'application Siges fournit par M. Blanc). Dans ce document les diagrammes sont en Mérise. Même si je n'ai pas appris ce concept à l'IUT, je le comprend car il est très proche d'UML donc pas de problèmes particuliers pour la lecture. Il faut juste faire attention car les cardinalités sont inversées.

Dans ce document, nous avons ajouté par la suite les flux de données. Pour chaque champ de formulaire à afficher, il a fallu effectuer des requêtes pour trouver d'où vient l'information dans la base de données. Pour cela, nous nous sommes connecté au serveur à l'aide du logiciel « client terminal serveur ».

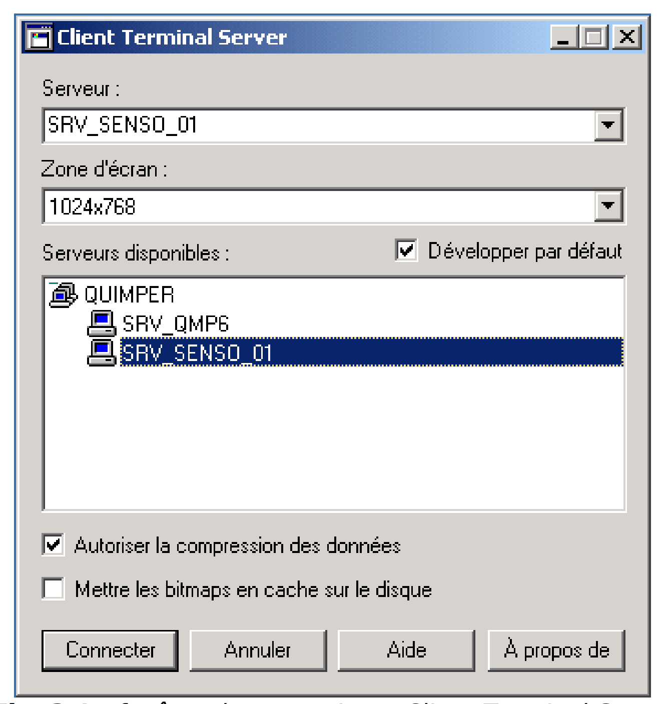



***Fig. 3.1 :** fenêtre de connexion – Client Terminal Server*

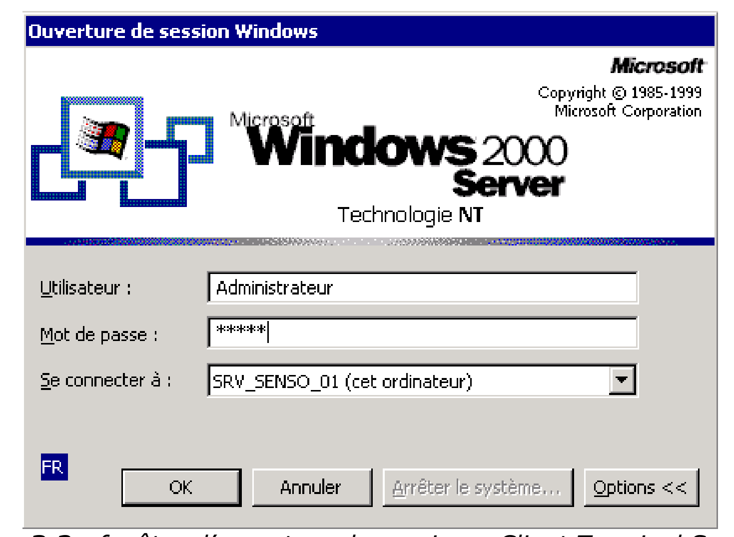



***Fig. 3.2 :** fenêtre d'ouverture de session – Client Terminal Server*





Puis nous avons utilisé le logiciel SQL Plus pour se connecter à la base de données.

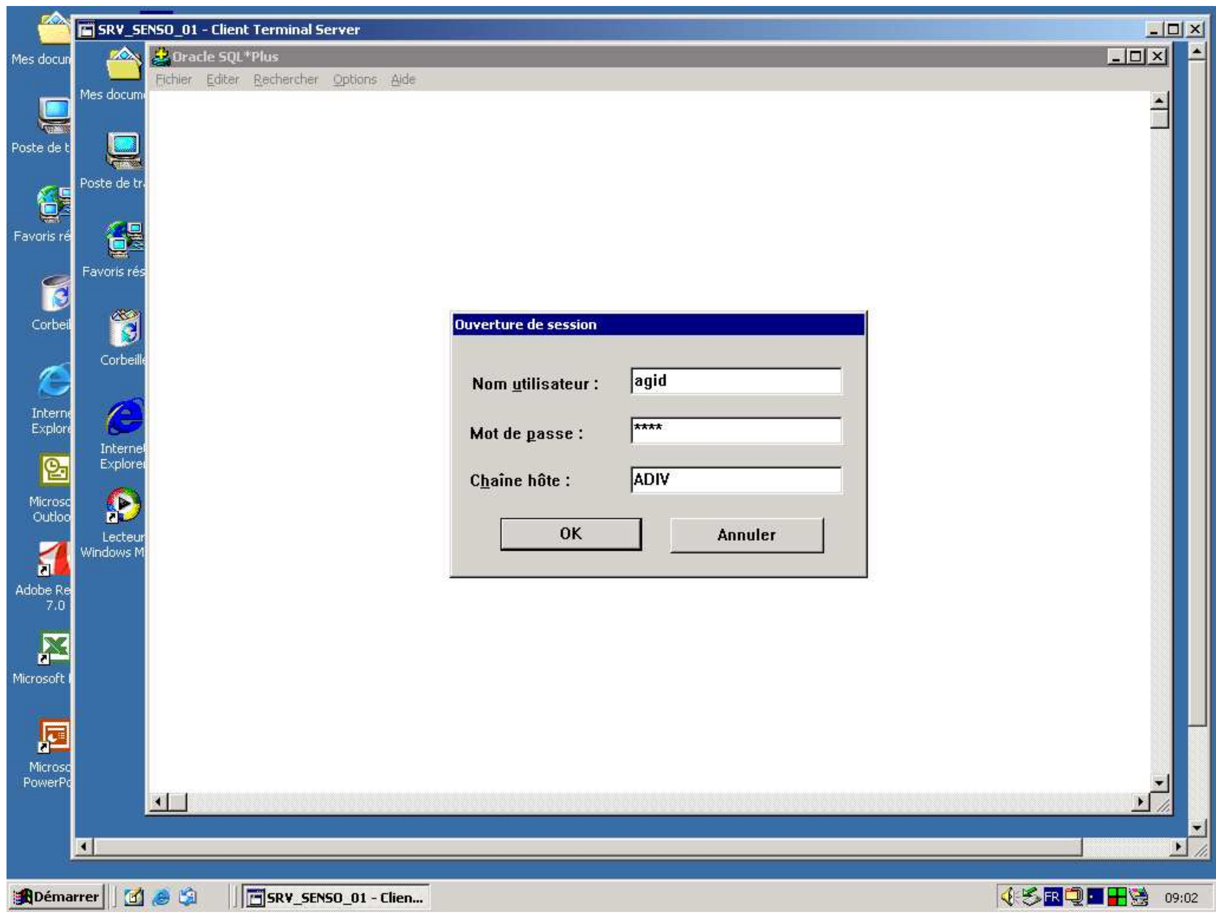



**Fig. 3.3 :** *ouverture d'une session dans SQL*Plus*





Voici un exemple de différentes requêtes effectuées dans le logiciel :

```
Oracle SQL*Plus
Fichier  Editer  Rechercher  Options  Aide
 ET_DATEPREV_FAX_INI                                 DATE
 ET_DATEPREV_COURRIER_INI                            DATE
 ET_PRORATA                                 NOT NULL VARCHAR2(1)
 ET_VALIDATION                              NOT NULL VARCHAR2(1)
 ET_DATEFACTURATION                                  DATE
 ET_ARCHIVE                                          VARCHAR2(20)
 ET_DATEGENPACHATS                                   DATE
 ET_FACTURATION                             NOT NULL VARCHAR2(1)
 ET_DATEDISPOPR                                      DATE

SQL> SELECT * FROM ETUDES WHERE ET_NUMERO='20040055';

ET_NUMER CO_ CO_ CO_ BA_CODE  SE_CODE  VA E E
-------- --- --- --- -------- -------- -- - -
ET_INTITULE_LONG                         ET_INTITULE_COU
---------------------------------------- ---------------
ET_OBJET_1
--------------------------------------------------------------------------------
ET_OBJET_2
--------------------------------------------------------------------------------
ET_CONTRAT_LONG                          ET_CONTRAT_COUR ET_DATE_ ET_DATE_
---------------------------------------- --------------- -------- --------
ET_DATEP ET_DATEP ET_DATER ET_DATER E ET_NFICH E E ET_COEFFRAISACC
-------- -------- -------- -------- - -------- - - ---------------
ET_FORFAITFRAISACC_FRF ET_FORFAITFRAISACC_EUR E ET_DATEP ET_DATEP E E ET_DATEF
---------------------- ---------------------- - -------- -------- - - --------
ET_ARCHIVE           ET_DATEG E ET_DATED
-------------------- -------- - --------
20040055 MF  VE  MF                       D E
SCAM01 PUREE AU LAIT 3X125G
TEST CONSO

PUREE AU LAIT 3X125G                                     15/11/04 24/10/05
                                    N          0 N
                                              N                   N 0
                     03/02/05 N


SQL> |
```

***Fig. 3.4 :** exemple de requête effectuée dans SQL*Plus*

Tout ce travail est disponible dans le dossier d'analyse et de conception en annexe.



### 1.1.3. Créer la base de test

En parallèle de la rédaction du document d'analyse et de conception, nous nous sommes occupés de l'installation de la base de test. Cette tâche a été effectuée dés le début du stage pour éviter un éventuel problème et commencer directement la phase de codage. Cela nous a pris tout de même trois semaines. Cette base a été installée sur un ordinateur qui va nous servir de serveur de test. Nous avons accès à cet ordinateur par le réseau intranet de l'entreprise.

### 1.1.4. Rédiger le document de début de stage

A l'aide du travail effectué pendant les deux premières semaines, j'ai pu facilement élaboré le document de début de stage que j'ai envoyé en fin de semaine 2 à mon responsable de stage M. Gilles Gaffiot.
Selon « le document de recommandations générales » distribué aux étudiants avant le stage, « l'objectif de ce document est de mesurer le degré d'investissement de l'étudiant dans son stage ». Je pense que ce document a bien été détaillé afin que M. Gaffiot puisse comprendre le sujet de départ du stage ainsi que les divers renseignements sur l'entreprise d'accueil.

## 1.2. Durant la phase de codage

### 1.2.1. Préparer le codage

Avant de passer au codage de l'application, nous avons installé, sur l'ordinateur qui nous sert de serveur, Apache 2.0 et PHP5, puis nous avons effectué la liaison entre les deux. Pour cela, nous avons configuré Apache et PHP en modifiant le fichier httpd.conf situé dans le dossier conf où est installé Apache (pour Apache) et en modifiant le fichier php.ini situé dans le répertoire où l'on a installé PHP.
Dans le fichier httpd.conf, nous avons définit en particulier le répertoire racine de l'application, les restrictions d'accès, etc.
Dans le fichier php.ini, nous avons définit comment seront géré les variables GET, POST, SESSION, quels sont les modules (.dll) à utiliser etc.

La liaison avec notre base de test Oracle a été faite automatiquement parce qu'Apache et la base se trouve sur le même poste. Mais on s'est aperçu que si ce n'était pas le cas, il aurait fallu configurer le fichier TNSNAMES.ora situé dans le dossier d'Oracle.

Lors de la préparation du codage nous avons écrit un manuel d'installation (annexe) qui décrit l'installation sur le serveur d'Apache et de PHP ainsi que la base Oracle (si base de test). Ce document pourra être réutilisé par d'autres personnes.

De plus, nous avons créé les différents dossiers tels qu'ils ont été définis dans le document d'analyse et de conception dans la rubrique « architecture interne de l'application ». Ces dossiers contiendront les sources PHP et fichiers de l'application. Par exemple, le dossier moteur contient les classes PHP qui gèrent le fonctionnement de l'application ou le dossier classes qui contient les classes PHP correspondantes aux tables de la base de données.

Pour toutes les manipulations à effectuer sur le serveur, il n'est pas forcément obligatoire d'aller sur la machine serveur. Par exemple, pour la création des dossiers nous nous sommes connectés à partir de notre machine à la machine qui nous sert de serveur à l'aide de la fenêtre Executer dans le menu démarrer et en tapant cette ligne de commande :

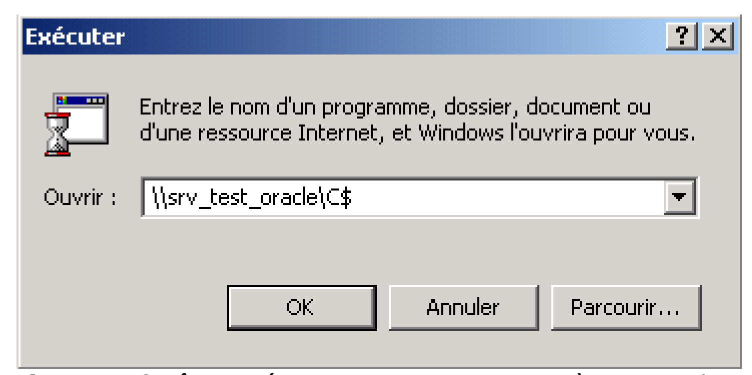



***Fig. 3.5 :** fenêtre Exécuter pour se connecter à une machine*

srv_test_oracle est le nom de la machine serveur.
C$ permet d'accéder au disque C de la machine avec une connexion persistante.

Une fenêtre demandant le login s'affiche. On tape les informations comme si on était sur la machine.

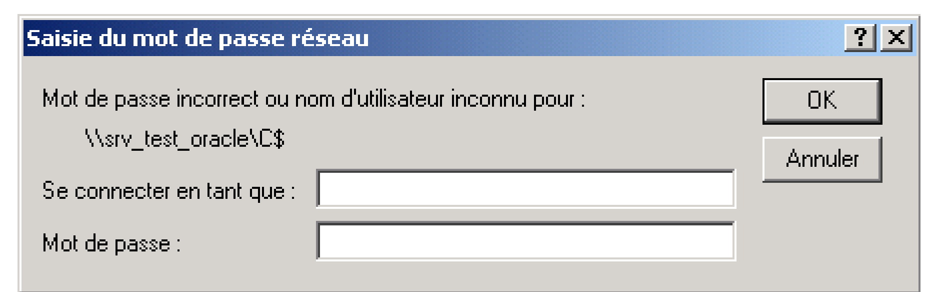



***Fig. 3.6 :** identification pour se connecter sur la machine*

Nous sommes ensuite connecté à l'ordinateur. Nous avons accès à tous les dossiers du disque C.

La préparation du codage s'est effectué lors de la troisième semaine.

### 1.2.2. Développer l'application

Après la préparation, nous avons pu passer au codage de l'application, dés la fin de la troisième semaine du stage.

Pour commencer, il a fallu élaborer une première page en HTML pour proposer un style à M. Blanc. Nous avons choisis le bleu et le blanc comme couleurs dominantes en lien avec le logo de l'Adria.

Comme ni moi ni Thibault ne connaissions comment gérer la connexion entre PHP et la base de données nous nous sommes renseignés sur le net. Nous avons ensuite créé une classe DataBaseManager pour la connexion puis une classe de test PHP utilisant la classe précédente pour vérifier le bon fonctionnement. On s'est rendu compte que PHP se lie très facilement à Oracle, aussi facilement qu'à MySQL d'ailleurs.

Nous avons ensuite réifié sous la forme objet les différentes entités considérées dans la base de données en prenant uniquement celles qui ont été identifié lors de la phase d'analyse (e.g. Etudes, Produits, Tiers etc.).

Pour que le code soit bien organisé et plus lisible nous avons recherché à organiser notre développement. Nous avons donc recherché sur le net des méthodes. Nous pensions utiliser des moteurs de templates. On en trouve un grand nombre sur le net tel que VTemplate ou FastTemplate mais ces moteurs ne nous ont pas convaincus. Nous avons donc préféré créer notre propre moteur et organiser les sources de l'application comme nous le souhaitions. L'organisation interne est expliquée dans la partie suivante.

La tâche suivante a été d'afficher la liste des fiches études (par page ou non) et de permettre d'effectuer une recherche par critères. Puis, nous avons créé toutes les pages en développant chaque formulaire. Le but a été dans un premier temps d'afficher les pages avec les champs remplis en fonction de la sélection d'une fiche étude. C'est seulement dans un deuxième temps que nous nous sommes occupés de l'enregistrement des modifications des données concernant une fiche donnée.

Nous nous sommes ensuite occupés de la configuration, de la gestion des collaborateurs (pour l'administrateur) et de la gestion de la connexion. Ces fonctions ont été codées en même parce qu'elles font toutes appel à un fichier xml. Pour gérer facilement l'enregistrement de données dans un fichier xml, nous avons utilisé les fonctions de la librairie DOM XML fournie par php. Nous utilisons un fichier xml car nous n'avons pas l'autorisation de toucher à la base de données.
Pour la connexion, nous avons été obligé de permettre l'enregistrement du login et du mot de passe de chaque collaborateur parce que nous n'avions pas le code (en Delphi) qui permet de décrypter les mots de passe enregistrés dans la base de données. Les utilisateurs de l'application que nous développons ont donc deux mots de passe qui peuvent être identiques ou non : un pour se connecter dans le logiciel Siges et un autre pour se connecter dans l'application web. Le mot de passe du collaborateur est enregistré crypté en md5 dans le fichier xml. La connexion est gérée par sessions.

Nous sommes ensuite passé à l'impression des pages. Pour cela, nous avons formaté le style css pour l'impression et pour chaque page nous avons fait en sorte que l'utilisateur puisse dérouler l'ensemble des différents sous-onglets pour pouvoir imprimer le détail complet d'un onglet. Nous sommes conscient que ce n'est pas la méthode la plus adaptée. Mais, par manque de temps (il ne nous restait que quelques jours), nous avons choisis la méthode la plus rapide à mettre en œuvre. Le problème est que ce formatage entraîne l'impression d'un grand nombre de pages, lorsqu'un onglet contient un grand nombre d'information. Pour palier à ce problème, l'utilisateur peut imprimer en 2 pages par feuille, recto verso.

Enfin, nous avons rédigé le manuel d'utilisation (en annexe).

### 1.2.3. Organisation des sources de l'application

Le schéma ci-dessous présente l'organisation interne de l'application.

./
index.php
new IncludeManager( $page )
./moteur/
IncludeManager
include( $incl->get_include() )
./pages/
$page (.php)
new FonctionDBManager()
new Formateur()
./templates/
./moteur/
$template (.tpl)
get
Formateur
new FonctionDBManager()
./classes/
./moteur/
les_classes (.php)
FonctionDBManager
require_once
new DataBaseManager()
./moteur/
DataBaseManager

***Fig 3.7 :*** *organisation interne de l'application*

Pour comprendre ce schéma, il faut déjà expliquer la légende.
Le texte en noir dans les encadrés correspond à un fichier (et/ou une classe donnée) ou à un groupe de classe (encadré « les_classes (.php)).
Le texte en vert indique où sont situés ces fichiers lorsque l'on part de la racine de l'application.
Le texte en rouge donne les actions effectuées par chacun des fichiers.





Que se passe-t-il lorsque l'on va à la page d'accueil de l'application ?
On tape l'adresse pour accéder à la page d'accueil c'est à dire http://srv_senso_01/index.php ou http://srv_senso_01/ (c'est la même chose).

La fenêtre de la page d'accueil s'affiche.

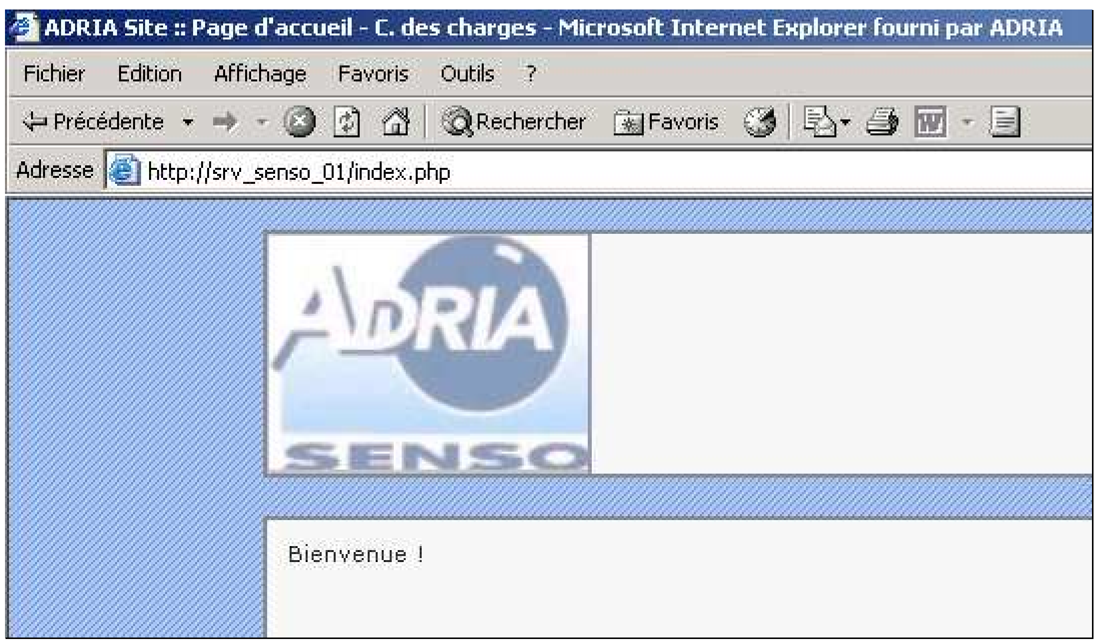



**Fig. 3.8 :** *page d'accueil*

index.php est en quelque sorte le lanceur de l'application.
Le code de ce fichier pourrait presque se tenir qu'à ces quelques lignes :

```
// Création d'un nouvel objet IncludeManager pour afficher une page
$incl = new IncludeManager($page);

// Affichage de la page
include($incl->get_include($_SESSION['myCode'], $_SESSION['pwd']));
```

Le fichier index.php s'occupe de créer un nouvel objet IncludeManager. On passe en paramètre le nom de la page à afficher que l'on récupère dans la barre d'adresse.





S'il n'y aucun paramètre « page » dans la barre d'adresse du navigateur IncludeManager prendra par défaut la page accueil.php. La méthode get_include() permet donc de renvoyer l'adresse de la page à afficher.

```
// contrôle du contenu demandé par la page
switch ($this->get_page()) {
    case "accueil": $include = "pages/accueil.php"; break;
    case "listeFiches": $include = "pages/listeFiches.php"; break;
    // etc.
    // [...]
    case "configuration": $include = "pages/conf.php"; break;
    default: $include = "pages/accueil.php";
}
```

Bien sûr, elle vérifie que l'utilisateur est bien connecté sinon on affiche par défaut la page d'accueil.

On arrive donc au stade où la page désirée est incluse.

Les pages s'occupent d'afficher ce qu'elles veulent en allant piocher dans les méthodes de la classe Formateur et/ou FonctionDBManager.
Toutes les fonctions de Formateur s'occupent de l'affichage. Une méthode particulière de Formateur est get_template(). Cette méthode va chercher un template par son nom dans le dossier templates et retourne son contenu sous forme d'une chaîne.

La classe FonctionDBManager offre une abstraction de la base de données. Cette classe offre par exemple des fonctions permettant :
- d'obtenir la liste des produits d'une étude,
- d'obtenir la liste des collaborateurs,
- d'obtenir une fiche étude (objet) en fonction de son numéro passé en paramètre,
- de modifier les informations concernant une étude,
- d'ajouter une nouvelle étape (date) concernant une sous-fiche,
- etc.

Cette classe créée a besoin de créer un objet DataBaseManager pour avoir accès à la base de données et pouvoir effectuer ses requêtes.

La plupart des fonctions de Formateur renvoient l'affichage d'une liste donnée telle que la liste fiches étude, la liste des produits d'une étude etc. Pour cela, Formateur se sert des fonctions de FonctionDBManager et formate l'affichage en HTML sous forme de tableau.

De plus, la page Index.php s'occupe de créer une session propre à chaque utilisateur et à chaque connexion à l'application. En effet, au début du fichier nous avons ajouté la fonction session_start() qui permet d'initialiser une session.
Pour s'en apercevoir, il suffit de cliquer sur le logo de l'Adria Senso qui possède un lien vers la page d'accueil. Bizarrement, on aura dans la barre d'adresse une url différente. La variable dite de session, PHPSESSID, a été ajoutée. Elle indique le nom du fichier qui a été créé uniquement pour l'utilisateur. Les fichiers de session sont créés dans un dossiers sessions sur le serveur. Si un autre utilisateur se connecte à l'application la variable PHPSESSID sera différente.




***Fig. 3.9 :** barre d'adresse et sessions*

### 1.2.4. Rencontrer mon tuteur de stage

Mon tuteur de stage à savoir M. Gilles Gaffiot est venu nous rendre visite dans l'entreprise le Mercredi 24 Mai. Cette rencontre s'est bien déroulée. A ce moment, nous étions arrivé à la fin du développement des différents onglets avec le remplissage des champs en fonction de la fiche étude.





## 1.3. Durant la phase de test

Durant cette phase, nous avons effectué les tests en mettant à jour le rapport de test de validation et de performance (annexe). Dans le même temps nous avons permis aux utilisateurs de Siges de l'Adria Senso de Quimper et de Toulouse de pouvoir utiliser l'application développée pour la tester. Ce test s'est effectué sur la base de test pour éviter d'avoir des problèmes avec la base réelle. Les utilisateurs n'ont pas trouvé de problèmes particuliers. En revanche, ils souhaitaient que soit ajoutée la duplication de dates dans le sous-test d'une fiche. Ce que nous avons fait.

M. Blanc nous a alors demandé d'envoyer le nécessaire au site de Clermont-Ferrand pour qu'il puisse installer l'application et la tester de leur coté. Nous avons donc envoyé le document de développement, le manuel d'utilisation, les sources de l'application, les fichiers de configuration de PHP (php.ini) et d'Apache (httpd.conf) ainsi que les logiciels d'installation de PHP5 et d'Apache2.0.

## 1.4. Le déploiement de l'application sur le serveur

Après la phase de test, nous avons pu déployer l'application sur le serveur de Quimper. Pour cela, nous avons installé PHP et Apache, copié, collé les deux fichiers de configuration php.ini, httpd.conf en les modifiant légérement pour que le serveur Apache fonctionne. Puis, nous avons modifié notre fichier de configuration config.php pour pouvoir avoir accès à la base réelle. Enfin, nous avons ajouté à la variable d'environnement Path le chemin d'accès à PHP et à Apache. Cette tâche nous a pris une matinée et s'est effectuée sans problème.

## 1.5. Et tout au long du stage…

Rédaction du rapport de stage.

# 2. L’aspect humain : une composante importante lors du stage

## 2.1 Organiser le travail en binôme

Puisque le stage doit s’effectuer en binôme (Thibault Le Reste et moi-même) il est important d’organiser le travail. Pour cela, le but est de séparer les tâches et de faire en sorte que l’on relit ensuite chacun le travail de l’autre pour éviter des erreurs éventuelles.
C’est surtout lors de la phase de codage que l’organisation est importante. En effet, il faut faire en sorte que l’on ne modifie pas le même fichier pour éviter de supprimer le travail de l’autre.
Par exemple, lorsque quelqu’un créé la classe Formateur, l’autre créé la classe FonctionDBManager.

## 2.2. Le suivi de M. Jean-Hugues Blanc très bien effectué

Pendant le stage M. Blanc venait nous voir ou nous allions le voir assez régulièrement. Le but était de faire une réunion avec lui au moins une fois par semaine pour lui indiquer l’état d’avancement du projet et pour qu’il nous dise ce qu’il souhaitait pour les jours suivants.
Je trouve que son suivi était idéal puisqu’il suivait notre travail régulièrement tout en nous laissant une grande autonomie.


RAPPORT de STAGE
45/59

# Chapitre 4

## Des résultats obtenus positifs







# 1. Les résultats obtenus concernant l'application web

Voici les fonctions qui ont été développées lors du stage :

- listage des fiches études par page (en fonction du module de l'utilisateur)
- listage de toutes les fiches études (en fonction du module de l'utilisateur)
- recherche des fiches études selon différents critères
- affichage des informations d'une fiche étude (Caractéristiques générales, tiers, produits, tests consommateurs, analyse)
- formater une page pour l'impression
- connexion à l'aide du login et du mot de passe
- configuration personnelle
- gestion des collaborateurs (par l'administrateur)

Concernant les performances, les tests sur le serveur ont démontré que l'affichage des différentes fenêtres de l'application est très rapide et/ou plus rapide que l'application Siges.

Voici ci-après, un aperçu de quelques fenêtre pour voir le résultat.

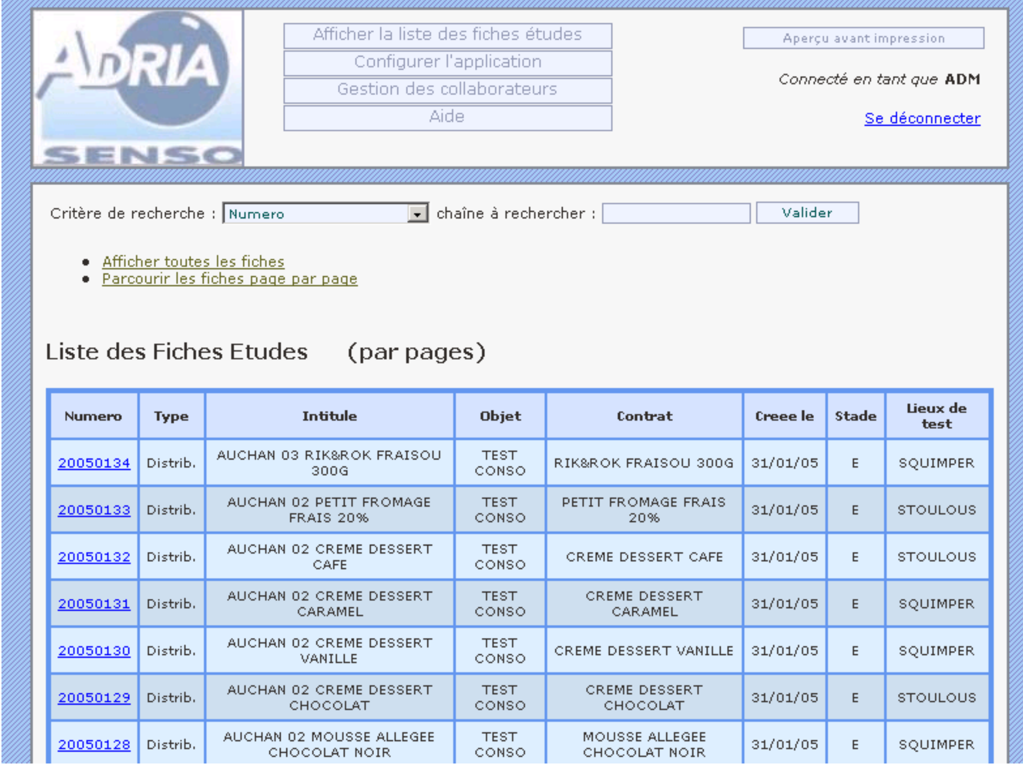



***Fig. 4.1 :** page d'affichage des fiches études (par page)*

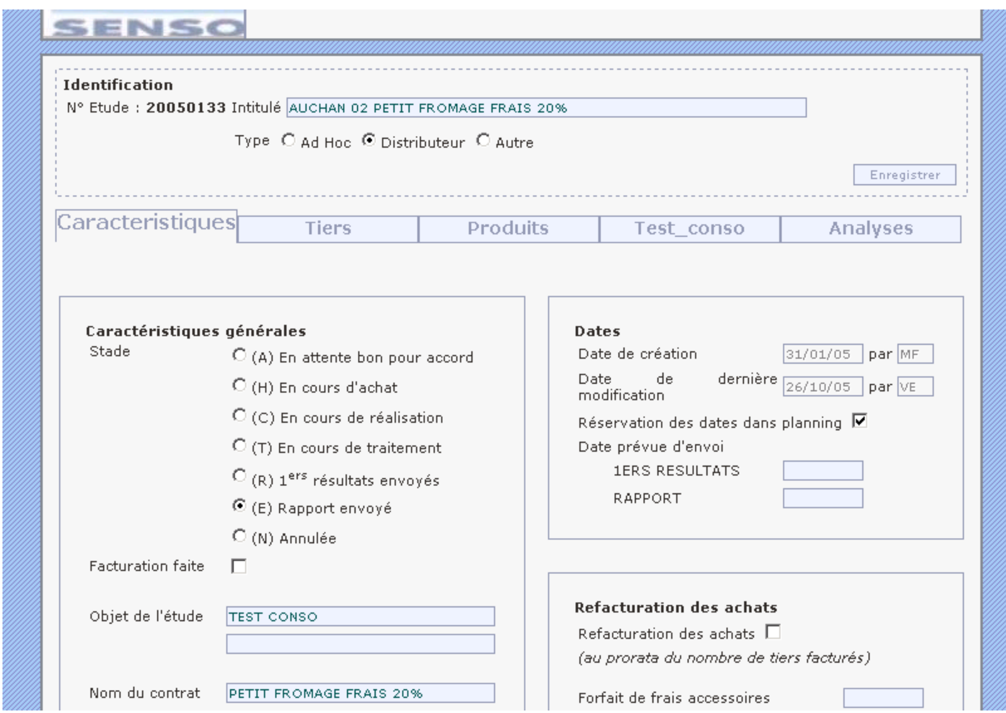



***Fig. 4.2 :** page de détail d'une fiche (onglet caractérisitques)*



# 2. Les problèmes rencontrés

Le principal problème rencontré a été de créer une copie de la base existante pour les tests lors du développement. Nous pensions que cette création serait effectué en deux semaines durant la phase de définition des besoins et l'analyse. Mais, la création d'une copie de la base nous a prit en fin de compte une semaine de plus. Pourtant dés la première semaine nous avions trouvé sur le net de nombreux sites expliquant clairement comment effectuer cette tâche. Nous avons donc installé Oracle sur mon ordinateur qui devait servir de serveur de test. Mais le problème rencontré était que lorsque l'on essayait de créer une nouvelle base, une erreur était générée et la création ne se faisait pas. Nous avons donc fait appel à tour de rôle à M. Aucouturier mais durant les jours où il cherchait à résoudre le problème, Nicolas a trouvé la cause. Pour cela, il a essayé tout simplement d'installer Oracle sur un autre ordinateur. Celui-ci possède Windows 2000 Server alors que mon ordinateur possède Windows 2000 Professionnel et apparemment Oracle n'accepte pas ce système. Pour le développement, nous avons donc eu un troisième ordinateur qui nous servait seulement de serveur comme si cet ordinateur était le serveur de Quimper et nos postes se situait dans une salle à Quimper ou à Toulouse.

Une autre difficulté rencontrée a été de trouver dans la base de données qui contient une soixantaine de table l'attribut qui correspond à chacun des champs recherchés. Nous y sommes parvenu mais ce travail nous a fait perdre un grand temps. Pour mieux comprendre, la démarche est expliquée au point 1.1.2 du chapitre 3.

Enfin, la gestion de l'impression n'est pas vraiment satisfaisante. L'utilisateur peut tout de même imprimer le détail d'une fiche étude. Mais, le problème est qu'il doit le faire onglet par onglet. De plus, dés que le sous-détail est grand ceci fait que nous avons un grand nombre de page. Ceci est causé par les sauts de page que nous avons été obligés d'ajouter pour ne pas avoir des détails coupés. L'idéal aurait été de proposer des pages d'impression en fonction de divers critères choisis par l'utilisateur. Par manque de temps nous n'avons donc pas pu utiliser cette méthode. En effet, il ne nous restait que deux ou trois jours à cause des problèmes rencontrés avant. C'est pourquoi je pense que si l'on doit dans le futur apporter une modification à l'application cela doit passer par cette gestion de l'impression.

Tous ces problèmes ont fait qu'il y a eu un écart par rapport aux prévisions.

Voici le planning que j'avais prévu en fin de deuxième semaine de stage.

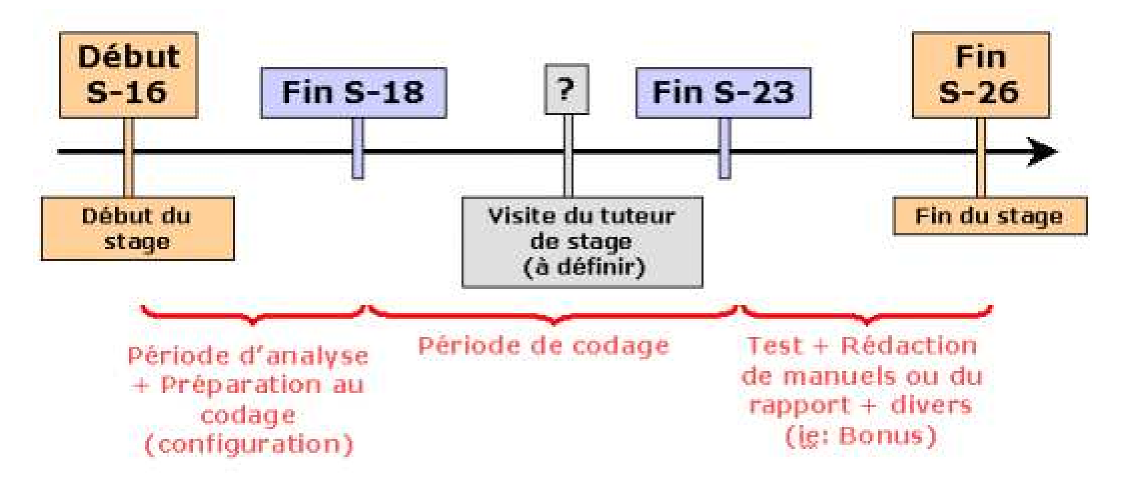



***Fig. 4.3 :*** *planning prévisionnel (fin de deuxième semaine)*

Et, voici le planning qui a été réellement effectué :

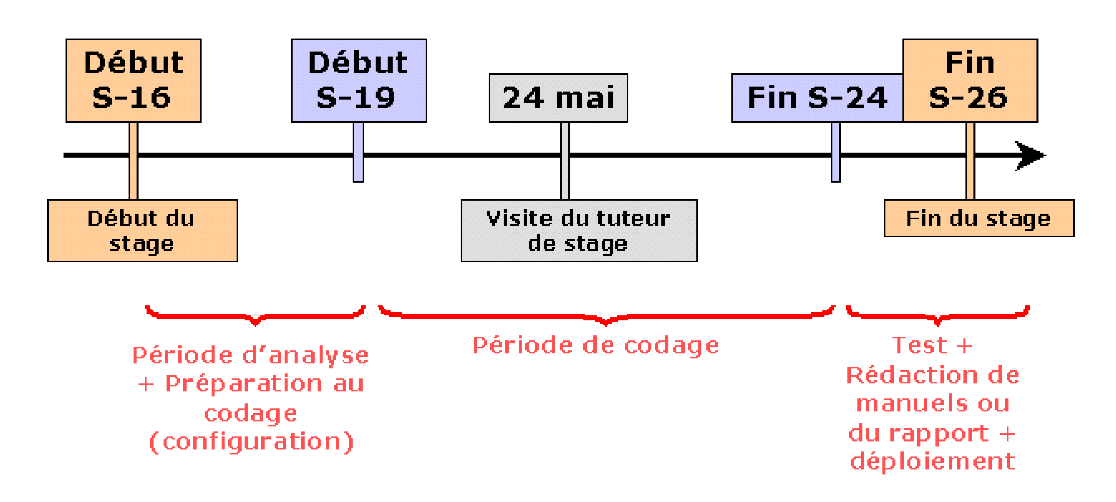



***Fig. 4.4 :*** *planning final*

On peut donc remarquer que la période de codage a été augmenter au détriment de la phase de test.







# 3. Les logiciels utilisés pour la résolution du problème

Voici les logiciels que nous avons utilisés et qui étaient mis à notre disposition dés le début du stage :

**Siges**
C'est le logiciel actuellement utilisé par l'ADRIA SENSO pour la gestion de ses fiches études.

**Oracle 8i**
Logiciel permettant la gestion de la base de données.

**Client Terminal Server**
Logiciel permettant d'avoir un accès à la machine serveur.

**SQL Plus**
Logiciel que nous avons beaucoup utilisé, permettant de se connecter à une base de données et d'effectuer des requêtes sur celle-ci.

**Internet Explorer**
Navigateur Internet permettant d'afficher des pages provenant d'Internet. Permet aussi d'afficher les pages de l'application que nous avons développée et d'effectuer les différents tests.

**Les outils Microsoft Office**
Ces logiciels m'ont permis de rédiger les différents manuels ainsi que le rapport de stage (avec Word) ou construire ma présentation (avec PowerPoint) pour la soutenance.

**Paint**
Logiciel pour la création graphique que ce soit pour la rédaction des rapports ou pour le développement de l'application.

Les logiciels suivants ont été installés durant le stage :

**Bouml** (http://bouml.free.fr/ | Gratuit)
Pour construire les diagrammes UML, nous ne pouvions pas utiliser Poseidon (comme à l'IUT) car ce logiciel est payant. Nous avons donc cherché un logiciel gratuit et choisit celui-ci.

**ConTEXT** (http://www.context.cx/ | Gratuit)
Logiciel pour développer l'application permettant la coloration syntaxique des langages que nous allons utiliser.

**PdfCreator** (http://sourceforge.net/projects/pdfcreator/ | Gratuit)
Ce logiciel permet de créer, fusionner des fichiers pdf à partir de fichier .ps. Cela permet en particulier de créer des pdf même si les dossiers et documents ont été écrit avec Word.

**APACHE (version 2.0.54)** (http://httpd.apache.org/download.cgi | Gratuit)
Permet de faire fonctionner l'application web. Il joue le rôle de serveur web.

**PHP (version 5.0.1)** (http://www.php.net/downloads.php | Gratuit)
Permet de lire des pages codées en PHP5 : programmation orienté objet (POO).
Pour cela, il faut disposer d'un serveur Apache et lier les deux.





# 4. Les enseignements réinvestis pour la résolution du problème

Les enseignements de l'IUT qui ont été réinvestis dans la résolution du sujet de stage ont été très nombreux et variés.
Concernant la phase d'analyse, j'ai utilisé les cours de Génie Logiciel pour concevoir le dossier contenant les différents diagrammes UML. Cette matière m'a aussi servit pour effectuer les nombreuses requêtes sur la base de données Oracle (SELECT, UPDATE, INSERT). De plus, le projet Computer Shop DB qui consistait à concevoir un site de commerce électronique est très en rapport avec le sujet puisqu'il a fallu utiliser PHP5 en lien avec une base de données.
Le cours « système et réseau » m'a aidé à faire fonctionner l'application web aussi bien à Toulouse qu'à Quimper en configurant par exemple le serveur Apache (fichier httpd.conf).
Les cours de programmation m'ont aidé à développer l'application du coté de la programmation orienté objet (POO) puisque PHP5 possède cet avantage.
Enfin, le projet de synthèse effectué durant la deuxième année m'a servit à organiser le stage et mieux gérer mon temps.



# Conclusion

Pour conclure, ce stage a été très complet. Je peux dire que le bilan est très positif. Au final, nous offrons tout de même 130 pages d'annexes et l'application possède environ 16000 lignes. L'application fonctionne parfaitement. A l'issue de la phase de test aucun problème particulier n'est ressortit.

Bien sûr, l'application développée ne reprend qu'une partie de l'application Siges. En effet, il reste d'autres fonctions propre à Siges qui n'ont pas été développées. Le but de ce stage était de démontrer qu'il était possible d'utiliser une autre méthode. Dans un premier temps, l'amélioration qui peut être apportée à l'application Web est la gestion de l'impression qui n'est pas assez satisfaisante. Dans le futur, l'ADRIA Senso pourra choisir de faire appel à une société informatique qui va redévelopper l'ensemble du logiciel Siges à l'aide de cette méthode. Elle pourra en profiter pour reconstruire la base de données car beaucoup de tables ou d'attributs ne sont pas utilisés et peuvent être supprimés. Dans l'autre sens, l'ADRIA Senso souhaitera peut-être en profiter pour ajouter de nouvelles fonctions et donc de nouvelles tables et de nouveaux attributs à la nouvelle base de données. J'ajoute qu'il faudra aussi en profiter pour relier les différents sites pour n'avoir qu'une seule base de données communes.

De mon coté, j'ai pu mettre en pratique les divers enseignements de l'IUT (Programmation, GL, Système et Réseaux, etc.). J'ajoute qu'en plus de cela, j'ai appris de nouvelles choses :

- installation et configuration complète d'un serveur Apache en lien avec PHP,
- création d'une nouvelle instance d'une base de données à l'aide d'un fichier DUMP,
- utilisation de PHP5 pour effectuer des requêtes de modification, d'insertion ou de sélection sur une base de données Oracle,
- création d'un moteur de site pour l'organisation de l'application en utilisant des templates.

Ce stage m'a offert une expérience professionnelle que je vais pouvoir ajouter dans mon CV. De plus, il m'a permis de me rendre compte que je suis très autonome et que je peut facilement travailler dans une petite équipe.
Ce stage a donc été pour moi un enrichissement personnel très grand qui me motive pour ma suite professionnelle. Je souhaite à présent intégrer une licence professionnel pour me spécialiser et m'insérer immédiatement dans le monde du travail.







# Table des illustrations







# Les annexes

- Document d'étude des langages serveurs
- Dossier d'analyse et de conception en vue de l'acquisition d'une application de gestion des fiches études
- Manuel d'installation
- Résultats des tests de fonctionnalités et de performance
- Manuel d'utilisation





# Résumé

Mon stage s'est effectué du 17 mai au 23 juin 2006 au sein de l'Adria Senso sur le site de Quimper. Cette société réalise des tests sensoriels sur divers produits du secteur agroalimentaire. Pour la résolution du problème, je travaillerai en binôme avec un autre stagiaire de l'IUT de Vannes, Thibault Le Reste. Nous devons redévelopper une partie d'un logiciel déjà existant et très utilisé dans l'entreprise appelé Siges. Ce redéveloppement doit permettre de fournir une application sous un autre format : le format web. Pour cela, j'utiliserais le langage PHP dans sa version 5.
Bien sûr, ce développement n'est pas sans problème. Il va falloir recopier la base de données Oracle, installer et configurer un serveur Apache, etc.
Ce rapport présente et analyse donc le contexte du stage, la résolution du problème et les résultats obtenus.

Mots Clés : Siges, PHP5, Apache, Oracle, application web, fiche étude

---

# Abstract

My training course was carried out from 2006, May 17 to June 23 in Adria Senso company on the site of Quimper. This company makes sensory tests on various products of the agroalimentary sector. For the resolution of the problem, I will work in binomial with another trainee of the IUT of Vannes, Thibault Le Reste. We must develop part of an already existing software and very used in the firm named Siges. This development must allow the firm to provide a software under another format : the web format. For that, I will use PHP language in his version 5.
Of course, this development is not without any problem. It will be necessary to recopy the Oracle data base, to install and configure an Apache server, etc.
Thus, this report introduces and analyzes the context of the training course, the resolution of the problem and the results obtained.

Keywords : Siges, PHP5, Apache, Oracle, web application, study card